Deug Mias-2, 2000-2001
Informatique
Institut Galilée

Projet

<u>Note</u>: Le projet est constitué de deux parties :

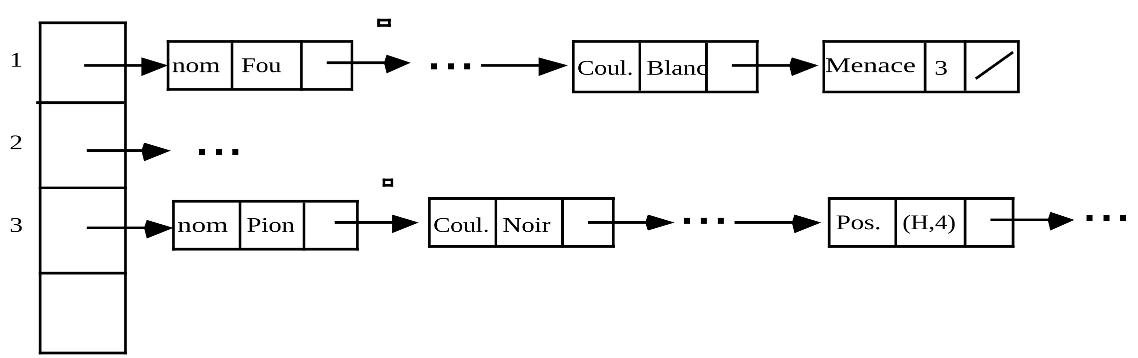
1) Un dossier où sont présentés le problème posé, les spécifications (que doit faire le logiciel ?), les choix de représentations, les solutions algorithmiques (comment et dans quelle mesure les spécifications sont-elles satisfaites?) ainsi que l'en tête et la définition des principales procédures et fonctions utilisées (qu'y a t'il en entrée et en sortie?, que fait exactement la procédure et sous quelles conditions?)

2) une disquette contenant le code-source (en pascal), le code-exécutable, un fichier texte README (le mode d'emploi mentionnant les conditions de compilation et d'utilisation s'il y a lieu). Le logiciel doit être exécutable sur une machine du département d'informatique (spécifiez laquelle si nécessaire) : le correcteur ne passera très probablement pas des heures à trouver une configuration permettant de lancer un programme..

Représentations relationnelles.

Nous représentons un ensemble d'entités (objets, individus, etc…) ainsi que leurs propriétés et les relations qu'ils ont avec d'autres entités (relations binaires) par une liste (ici un tableau) dont chaque élément représente une entité, et est lui-même une liste de couples (attribut valeur). Un attribut associé à une relation avec une autre entité a comme valeur une référence (ici un entier) vers cette seconde entité.

<u>Exemple</u>: Une position aux échecs est représentée par un ensemble de pièces. Une pièce



p1 est représentée par son type (p1 est un Fou), sa couleur (p1 est Blanc), ses relations avec d'autres pièces (p1 menace p2, p1 est menacé par p2, p1 est en échec, etc..): ici un fou blanc menace un pion noir qui est sur la case (H,4).

Attention, une pièce peut en menacer plusieurs. D'autre part il n'est pas nécessaire dans cette représentation que pour chaque entité (pièce) toutes les propriétés et relations aient un sens (et donc apparaissent dans la liste des éléments représentant une pièce) : par exemple le Roi et la Reine sont les seules pièces qui peuvent être mises en échec .

Le but du projet est de créer une telle base de connaissance, de la mettre à jour lorsqu'une nouvelle information modifie la situation, et de pouvoir répondre a des requêtes comme par exemple "Pièces menaçant le Pion en E4 ?". Ces requêtes peuvent être exprimées simplement de manière interactive.

Le domaine est laissé au choix des auteurs. Ce peut être la position d'un jeu comme dans l'exemple ci-dessus, ou tout autre domaine dans lequel il est intéressant de représenter les relations et les propriétés d'un groupe d'entités. Soyez originaux : évitez le jeu d'échecs, (sauf si vous êtes un passionné du jeu).

Il est conseillé d'être attentif à la création et à la sauvegarde de la base : on ne peut écrire sur la sortie (fichier) des adresses, il faut donc trouver une représentation "externe" et écrire les procédures de traduction de la représentation interne (avec les pointeurs) à la représentation externe et vice versa. Le plus simple est d'utiliser comme représentation externe un fichier texte donc chaque ligne contient une suite d'enregistrements dans lesquels le champ "suivant" est omis.

Remarque: Le Pascal imposant le typage des pointeurs vous aurez ici besoin d'enregistrements variables pour chaîner des éléments dont un champ donné peut être de plusieurs types différents selon l'élément. Un exemple d'utilisation est donné ci-dessous en Pascal.

```
Program  Variations (input, output);
Type
   ltype = ( entier, chaine);
   element = record
     variant : ltype;
     next : ^element;
     case ltype of
       entier: (ValEntier : Integer);
       chaine: (ValChaine :  String);
   end;
Var
  pElement1,   pElement2:^element;
Begin
  new(pElement1);
  pElement1^.variant :=chaine;
  pElement1^.ValChaine :='toto';
  new(pElement2);
  pElement2^.variant :=entier;
  pElement2^.ValEntier :=2;
  pElement2^.next :=nil;

  pElement1^.next :=pElement2
   (* l'intérêt  du champ variant est ici de permettre au
programmeur de savoir de quelle variante de valeur il s'agit
*)
```

Note: Les projets sont en principe faits en binôme. Si le projet est particulièrement ambitieux, il peut être présenté par trois ou exceptionnellement quatre étudiants.